EAComp
Encontro Acadêmico de Computação

VI Encontro Acadêmico de Computação - UFMA
 ISSN: 2594-486X

Desenvolvimento de um time de futebol robótico para o RoboCup 2D

Rodrigo Mendes Garcês[1],Jamesson Amaral Gomes[1], Geraldo Braz Júnior[1], Alexandre César Muniz de Oliveira[1], Luciano Reis Coutinho[1], Paulo Rogério de Almeida Ribeiro[1]

[1]Programa de Educação Tutorial
do curso de Ciência da Computacão
Av. dos Portugueses, 1966, Campus do Bacanga
65065-545 – São Luís – MA – Brasil

{rmgarces1,jamesson.ag}@gmail.com, {geraldo.braz,alexandre.cesar}@ufma.br,

lrc@deinf.ufma.br, paulo.ribeiro@ecp.ufma.br

Resumo. *Este trabalho descreve a simulação em duas dimensões do futebol de robôs do RoboCup, detalhando a estrutura do ambiente de simulação e desenvolvimento do futebol robótico, objetivando incentivar a criação de times e estratégias inteligentes por estudantes e pesquisadores da área. A partir da escolha e estudo do time base UVA Trilearn, foram desenvolvidas três estratégias básicas (Chutar ao Gol, Chutar de acordo com a coordenada do jogador e Drible) usando a linguagem de programação C++. A combinação destes possibilitou a criação do primeiro time de futebol de robôs simulado na UFMA. Deseja-se implementar futuramente um framework que possa abstrair comandos de baixo nível, objetivando a criação e teste de novos algoritmos, como sistemas multiagente, redes neurais, além de simular a resposta de um algoritmo autônomo a estímulos e percepções sensoriais, sem a necessidade de compreender todas as camadas de abstração e comunicação da plataforma. Embora o time ainda esteja em um estado inicial, já se tem mostrado promissor.*
Palavras-chave*: RoboCup, Futebol Robótico, UVA Trilearn.*

1. Introdução

Criada em 1996 [Kitano et al. 1997], a RoboCup fornece um problema padrão, o Futebol de Robôs. A comunidade internacional RoboCup promove o desenvolvimento de robôs inteligentes, definindo e executando competições que são usadas por cientistas e estudantes de todo o mundo para fins de teste e aperfeiçoamento de seus robôs em cenários o mais próximos da realidade possível. Tal competição se divide em várias subcategorias, sendo a simulação 2D a abordada neste artigo. E várias equipes disponibilizam parte de seus códigos, a fim de abstrair as implementações de baixo nível, recebendo a denominação de time base.

Este trabalho tem como objetivo a criação de um time de futebol robótico na plataforma RoboCup e desenvolver, ao final deste, um *framework* robótico para a referida plataforma, reduzindo a complexidade dos códigos e o conhecimento mínimo necessário de programação e plataforma em si. Foram implementados comandos de alto nível, como: procurar os adversários, roubar a bola (interceptar a bola sem cometer uma penalidade e com baixo custo de energia), reorganizar a formação do time durante o jogo, dentre outros.

O restante deste trabalho aborda, de forma resumida, os problemas e dificuldades encontradas no desenvolvimento do time inicial, bem como uma revisão bibliográfica utilizada para o desenvolvimento deste projeto de pesquisa.

2. Proposta Metodológica

A metodologia proposta para a construção do *framework* passou pela configuração do ambiente de simulação, levantamento de times bases relacionados e a escolha de um deles, extensão dos algoritmos através da implementação de 3 estratégias iniciais e avaliação dos testes.

A etapa de configuração consiste basicamente em colocar o servidor de simulação em funcionamento. O *RoboCup Soccer Simulator Server*, que realiza a simulação das partidas, sendo padrão na plataforma. Logo uma vez configurado, passa-se à fase de implementação dos algoritmos. Por esse motivo, este trabalho não aborda temas a respeito da configuração do *SoccerServer*. Cada uma das demais etapas é tratada com mais detalhes nas subseções seguintes.

3. Levantamento de Times Base

O time-base é uma plataforma arquitetada para abstrair códigos de baixo nível e auxiliar outras equipes, embora o seu uso seja opcional, visto que cada equipe pode desenvolver sua própria estrutura. Esta seção objetiva apresentar os times base (Helios Base (Agent 2D), UVA Trilearn, WrightEagle Base, Bahia 2D) mais utilizados no cenário mundial e nacional apontando suas principais características.

O time Helios [Akiyama and Nakashima 2013], de origem japonesa, disputa a *RoboCup Simulation League* desde o ano 2000, e disponibiliza parte de seus códigos (chamados de Agent2D) com a intenção de contribuir com outras equipes e com a sociedade acadêmica. É desenvolvido em C++, sendo atualizado com certa frequência. Possui ampla cobertura com relação às habilidades básicas, o que permite ao desenvolvedor centrar seu trabalho na construção de habilidades mais complexas.

O time WrightEagle [Li et al. 2015], da China, disputa a RoboCup Simulation desde o ano 1999 e obteve 3 vitórias consecutivas (2012-2015), ganhou duas vezes o campeonato mundial, e disponibiliza parte de seus códigos para a comunidade. Este *framework* possui um comportamento básico para a tomada de decisão e muitas implementações baixo nível. É bem modularizado sendo seu código separado por comportamento (Behavior) e por ações (Action). Utiliza a linguagem de programação C++.

A arquitetura adotada pelo BAHIA 2D [Silva and Simões 2009] é dividida em três camadas inter-relacionadas: nível reativo, instintivo, cognitivo. O reativo tem como finalidade executar ações, o instintivo toma decisões e o cognitivo consiste na camada inteligente, implementada através de redes neurais. Possui os métodos importados do UVA Trilearn e alguns do próprio Bahia2D, desenvolvidos em linguagem C++.

3.1. UVA Trilearn Base

Um outro time base é o UVA Trilearn [de Boer and Kok 2002]. Este foi escolhido por ser simples e apresentar documentação consistente. Apresenta como maiores

Copyright (c) EAComp. ISSN: 2594-486X

contribuições uma arquitetura de três camadas capaz de realizar o processamento paralelo dos dados, além de um esquema flexível de sincronização do ambiente, métodos eficazes para localização de objetos, dentre outros. Esta arquitetura permite que tarefas complexas possam ser decompostas em várias sub-tarefas simples. Há dois parâmetros que merecem destaque neste trabalho: um é o *kickable_margin*, que é a área em que o jogador consegue chutar a bola (algo em torno de 0.7m, dependendo do jogador.). O outro parâmetro é o *posXField*, que é a coordenada x da posição atual do jogador. Ambos os parâmetros serão utilizados na sessão seguinte. A arquitetura de três camadas é apresentada na Figura 1.

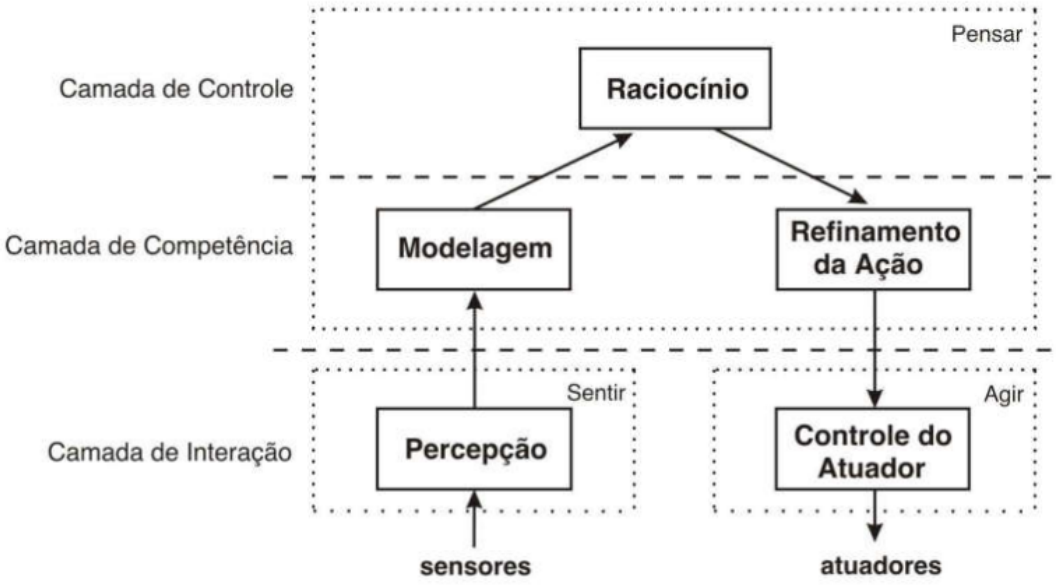

Figura 1. Arquitetura do time UvA TriLearn

Cada classe é melhor detalhada a seguir:

- Conexão: Cria uma conexão com o servidor, via *socket UDP*, provida de métodos para enviar e receber mensagens;
- Controle dos sensores: Processa as mensagens recebidas do servidor, particiona e envia as informações extraídas para o modelo de mundo (*world model*);
- Modelo de mundo (*world model*): Cria a representação atual do jogo, do ponto de vista observado por cada agente, sendo responsável por gerenciar todo o simulador;
- Objeto: Contém as informações sobre todos os objetos da simulação, tais como as demarcações do campo, os jogadores, a bola e a posição do gol;
- Formações: Contém as informações sobre as possíveis formações do time e os métodos para determinar as posições estratégicas de cada jogador;
- Configuração do jogador: Contém os valores dos parâmetros do jogador, tal como a influência para resolver algum processo do agente e de definir métodos para receber e atualizar esses valores;
- Jogador: Contém métodos para tomada de decisão, de modo a escolher a melhor opção em uma determinada situação;
- Jogador básico: Contém todas as informações necessárias para um perfil de jogador.

O time UVA Trilearn possui 3 formações táticas, *(3-4-3), (4-4-2)* e *(4-3-3)*. O UVA Trilearn utiliza a formação tática *(4-3-3)*, que corresponde em separar os jogadores ao longo do campo em áreas de atuação. No caso, os quatro primeiros jogadores atuarão como zagueiros, os três outros jogadores atuarão como meio campo e os últimos três jogadores atuarão como atacantes.

3.2. Estratégias Implementadas

Com a finalidade de implementar o time básico da UFMA, a partir do UVA Trilearn Base foram implementados algoritmos que extendem a base na forma de 3 estratégias de formação tática.

Na estratégia 1, o jogador chuta a bola com força máxima sempre que a bola está dentro da área de chute do jogador (quando a distância dela até o jogador é menor ou igual ao *kickable_margin*). Optou-se por chutar em um dos cantos do gol, dependendo do ciclo corrente da partida.

Na estratégia 2, identifica-se primeiramente a posição atual do jogador. Se a bola está em sua área de chute, verifica a sua posição em campo (ataque ou defesa). Estando no campo de ataque, na entrada da grande área e $x < 35$, então fará um lançamento para a coordenada $(posXField - 15)$, sendo que $posXField = 40$ e este valor fora selecionado por corresponder a entrada da pequena área , uma vez que a coordenada $y = -15$ é negativa o jogador realizará um passe para a posição superior do campo com velocidade de chute de 3m/ciclo, isto é, a parte de cima do campo de ataque, objetivando que o jogador mais próximo desse ponto receba a bola e chute ao gol.

Na estratégia 3, um dos jogadores deve carregar a bola driblando até a entrada da grande área, região onde a coordenada $x > 35$, e chutar em seguida em direção ao gol. Se o jogador estiver antes da posição $x = 35$, irá carregar a bola em um ângulo = $0°$, ou seja, para frente e driblando de forma a somente conduzir a bola. Quando passar da posição $x = 35$ chutará a bola para o gol assim como na Estratégia I.

4. Resultados

Esta seção apresenta os resultados de cada uma das 3 estratégias aplicadas ao time UVA Trilearn, em 3 partidas realizadas contra o time do Agente2D, visando analisar a atuação do time, considerando os pontos positivos e negativos obtidos ao acréscimo de cada estratégia. Adicionalmente, foram realizadas partidas entre o próprio UVA Trilearn com versões de estratégias diferentes, confrontando-se as estratégias II e III. As partidas duraram 2 tempos de 5 minutos cada, totalizando 10 minutos por partida.

4.1. Estratégia I: Chutar ao Gol

A Tabela 1 demonstra resultados obtidos com a estratégia I de chutar em direção ao ao gol, com potência máxima *(kick maximal)*.

Durante o 1º tempo foram marcados 0 gols e sofridos 9. No 2º tempo com a diminuição de *stamina* por parte de alguns jogadores do UVA Trilearn, devido ao chute com potência máxima, notou-se uma dificuldade para esse jogadores cansados interceptarem a bola, visando bloquear os adversários, assim mais 14 gols foram sofridos pelo time proposto, totalizando 23 gols, sendo que nenhum gol foi marcado utilizando-se a estratégia I, evidenciando-se a necessidade do desenvolvimento de estratégias mais refinadas para um placar mais equilibrado.

Tabela 1. Resultados da Estratégia I da partida UVA I x Agent2D.

UVA I x Agent2D	Gols Marcados	Gols Sofridos	Finalizações	Defesas
1º Tempo	0	9	0	0
2º Tempo	0	14	0	0
Total	0	23	0	0

4.2. Estratégia II: Chutar de Acordo com a Coordenada do Jogador

A Tabela 2 demonstra os resultados obtidos com a estratégia II, que além chutar ao gol utiliza a coordenada do jogador que vai realizar o passe ou o chute. Durante o 1º tempo foram marcados 0 gols e sofridos 12 pelo time proposto, apresentando um aumento de gols sofridos no 1º ciclo – quando comparado a estratégia I – o que pode ser explicado devido a maior ofensividade obtida com a estratégia II. No 2º tempo sofreu-se 11 gols totalizando 23 gols, sendo marcados 0 gols nos 2 tempos com a estratégia II, mostrando-se apenas uma melhora no 2º tempo em relação a Tabela 1 em que a estratégia I é aplicada.

Tabela 2. Resultados da Estratégia II da partida UVA II x Agent2D.

UVA II x Agent2D	Gols Marcados	Gols Sofridos	Finalizações	Defesas
1º Tempo	0	12	0	0
2º Tempo	0	11	0	0
Total	0	23	0	0

4.3. Estratégia III: Drible

A Tabela 3 ilustra os resultados obtidos com a estratégia III. Na estratégia III aplica-se a tentativa de drible, onde o jogador visa desviar do jogador adversário.

Tabela 3. Resultados da Estratégia III da partida UVA III x Agent2D.

UVA III x Agent2D	Gols Marcados	Gols Sofridos	Finalizações	Defesas
1º Tempo	0	13	1	0
2º Tempo	0	9	0	1
Total	0	22	1	1

A estratégia III aplicada na partida diminuiu o total de gols sofridos em relação as estratégias I e II, passando de 23 para 22 gols sofridos, assim como evidenciou finalizações e defesas.

4.4. Estratégia II e III Aplicadas em uma partida do UVA II x UVA III

Com as três estratégias propostas objetivou-se dar maior ofensividade, entretanto, percebeu-se que as partidas contra times de elevada capacidade técnica - nomeadamente Agent2D – não oferecem métricas satisfatórias para efeitos comparativos. Portanto, optou-se em realizar novas partidas utilizando-se somente o time proposto, ou seja, um time com a estratégia II e o outro time com a estratégia III, sendo seus resultados apresentados na tabela 4:

A Tabela 4 apresenta os resultados da partida do UVA Trilearn com as estratégias II e III confrontadas em uma partida de 2 tempos. Os resultados demonstram a melhor

Tabela 4. Resultados da Estratégia II e III da partida 1 UVA II x UVA III.

UVA IIIxAgent2D	Gols Marcados	Gols Sofridos	Finalizações	Defesas
1º Tempo	2	0	17	9
2º Tempo	1	1	10	14
Total	3	1	27	23

capacidade ofensiva dos times neste cenário, tanto em alcançar o campo adversário, como no número de defesas. Os resultados da Tabela 4 demonstram a vitória do time UVA II com a estratégia II por 3 x 1 em relação ao UVA III que utiliza a estratégia III. Sendo realizadas 17 finalizações e 9 defesas no 1º tempo e 10 finalizações e 14 defesas no 2º tempo, totalizando 27 finalizações e 23 defesas nos 2 tempos pelo time UVA II.

5. Conclusão

O *framework* demonstrou-se simples, embora o código da plataforma seja bastante extenso e altamente acoplado, o que dificulta bastante o entendimento inicial. A extensa documentação do time UVA Trilearn facilitou de maneira significativa a compreensão do funcionamento da plataforma, permitindo observar, de maneira simplificada, todo o funcionamento do simulador, tal como: A comunicação entre as classes/ objetos componentes do sistema, o processo de aquisição de dados do ambiente e o processo de tomada de decisões.

O time desenvolvido na plataforma ainda encontra-se em estágio inicial, mas tem demonstrado alguns resultados promissores. Tendo em vista o nível do time Agent2D, pode-se considerar que os resultados obtidos foram promissores, visto que obteve-se uma notória evolução no desempenho do time, embora ainda esteja distante de alcançar o nível de um time profissional. Destaca-se o goleiro, que não recebeu nenhum tipo de estratégia inteligente até o momento e ainda precisa ser aperfeiçoado.

Como trabalhos futuros, visa-se estudar mais profundamente o código-base dos principais times base e dos principais algoritmos utilizados por estes, a fim de criar estratégias mais elaboradas, podendo utilizar sistemas multiagentes, processamento de sinal, otimização, dentre outros.

Referências

Akiyama, H. and Nakashima, T. (2013). Helios base: An open source package for the robocup soccer 2d simulation. In *Robot Soccer World Cup*, pages 528–535. Springer.

de Boer, R. and Kok, J. (2002). *The incremental development of a synthetic multi-agent system: The uva trilearn 2001 robotic soccer simulation team*. PhD thesis, Master's thesis, University of Amsterdam, The Netherlands.

Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., and Osawa, E. (1997). Robocup: The robot world cup initiative. In *Proceedings of the first international conference on Autonomous agents*, pages 340–347. ACM.

Li, X., Chen, R., and Chen, X. (2015). Wrighteagle 2d soccer simulation team description 2015.

Silva, L. and Simões, M. A. (2009). Uma estratégia para seleção de agentes heterogênos para o futebol de robôs simulado.